

# GITLAB CI FOR DUMMIES

---

Dummy = Jon Mason  
Dummy != Bruce Ashfield (debatable)

# Dummies

Jon Mason

- Arm (very smart people)
- OpenEmbedded Board Member
- Co-maintainer of meta-arm layer

Bruce Ashfield

- Xilinx
- Maintainer of Yocto Project kernel, meta-virtualization layer

## Note:

You are not a dummy. I am a dummy. This took way longer that it should have, mostly because...I am a dummy. Learn from my mistakes to speed up your deployment and not be a dummy.



# What is Gitlab?

Web based source code management (SCM), similar to github. As the name implies, it is used for git based source code.

# Why Gitlab?

- Free
- Open Source
- Has robust features like wikispaces and...CI/CD

# What is this CI/CD?

Continuous integration (CI) is the practice of merging all developers' working copies to a shared mainline

Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, without doing so manually

Put simply, it is an extremely frequent building and testing of software patches as part of the process of integrating them into a shared tree

# Why is this useful for open source projects?

Maintainers of projects can get dozens (possibly hundreds) of patches daily. Identifying and removing faulty ones prior to releasing them to the public can avoid development issues with the wider developer community.

This process enables patches to be built and tested prior to integration, finding bugs BEFORE they break other developers of the shared code.

# Drawbacks

- In a perfect world, we would have enough computing resources to compile and test every patch, weeding out the problematic ones as they arrive.
- Unfortunately, this isn't feasible, as each CI/CD run can take hours and dozens (perhaps hundreds) of patches arrive daily. So, they must be tested in batches. This increasing the difficulty of determining faulty patches and requiring more manual intervention.
- For example, there were 97 patches pulled into poky between 05/10-05/17/21. If we compiled each one, for each qemu machine, and it only took an hour (which it takes way more), it would add up to 1067 ( $97 * 11 * 1$ ). (note: 168 hours/week)



# Gitlab CI

GitLab CI/CD is a tool built into GitLab for software development

Free (if you run everything on your own hardware, more about this later)

# Alternatives to Gitlab CI

- Github workflow
- Jenkins and LAVA
- Many others

# Current Users of Gitlab CI in YP/OE

- Meta-arm
- Meta-security
- WIP
  - Meta-zephyr (personally maintained)
  - Yocto Project kernel releases

Pipelines · Jon Mason / meta-arm

←

→

↺

🔒

gitlab.com/jonmason00/meta-arm/-/pipelines

GitLab

ProjectsGroupsSnippetsHelp

Search or jump to...

Sign in / Register

M

meta-arm

Project overview

Repository

Issues0

Merge requests0

Requirements

CI/CD

Pipelines

Jobs

Schedules

Test Cases

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Collapse sidebar

Jon Mason · meta-arm · Pipelines

All314

Finished

Branches

Tags

Filter pipelines

Status	Pipeline	Triggerer	Commit	Stages	Duration	
<div>failed</div>	<div>#308354374</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 master → 2b887e62</div> <div>🌐 arm/linux-yocto: additional ...</div>	<div>✓</div> <div>✓</div> <div>✗</div>	<div>🕒 00:55:58</div> <div>📅 8 hours ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>passed</div>	<div>#308035538</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 gatesgarth → 5fc1de74</div> <div>👤 Cl: fix preferred version of a...</div>	<div>✓</div> <div>✓</div> <div>!</div>	<div>🕒 01:32:43</div> <div>📅 18 hours ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>passed</div>	<div>#308022903</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 dunfell → d11487b6</div> <div>🌐 Cl: use public KAS image</div>	<div>✓</div> <div>✓</div> <div>✓</div>	<div>🕒 05:07:51</div> <div>📅 19 hours ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>passed</div>	<div>#308014717</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 hardknott → 7ca13b4f</div> <div>🌐 arm/linux-yocto: add missin...</div>	<div>✓</div> <div>✓</div> <div>✓</div>	<div>🕒 12:26:05</div> <div>📅 17 hours ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>failed</div>	<div>#307697733</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 master → 2b887e62</div> <div>🌐 arm/linux-yocto: additional ...</div>	<div>✓</div> <div>✓</div> <div>✗</div>	<div>🕒 09:56:19</div> <div>📅 1 day ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>passed</div>	<div>#307127891</div> <div>Scheduled</div> <div>latest</div>		<div>🔧 master → 2b887e62</div> <div>🌐 arm/linux-yocto: additional ...</div>	<div>✓</div> <div>✓</div> <div>✓</div>	<div>🕒 01:45:17</div> <div>📅 3 days ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>failed</div>	<div>#306784657</div> <div>latest</div>		<div>🔧 downstream → 1e738bb6</div> <div>🌐 arm/trusted-firmware-a: upda...</div>	<div>✓</div> <div>✓</div> <div>✓</div> <div>✗</div>	<div>🕒 00:49:24</div> <div>📅 3 days ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>
<div>failed</div>	<div>#306732776</div>		<div>🔧 downstream → 698be477</div> <div>🌐 arm/trusted-firmware-a: upda...</div>	<div>✓</div> <div>✓</div> <div>✗</div> <div>✗</div>	<div>🕒 00:34:19</div> <div>📅 4 days ago</div>	<div>▶</div> <div>▼</div> <div>⋮</div>

+

Pipeline · Jon Mason / meta-arm

←

→

↺

🔒

gitlab.com/jonmason00/meta-arm/-/pipelines/308354374

GitLab

ProjectsGroupsSnippetsHelp

Search or jump to...

🔍🔔👤Sign in / Register

M

meta-arm

🏠Project overview

📁Repository

📄Issues0

🔀Merge requests0

⚙️Requirements

🔗CI/CD

Pipelines

Jobs

Schedules

Test Cases

🔧Operations

📦Packages & Registries

📊Analytics

📖Wiki

✂️Snippets

👤Members

⏮️Collapse sidebar

Jon Mason > meta-arm > Pipelines > #308354374

❌failed

Pipeline #308354374 triggered 9 hours ago by Jon Mason

Retry

arm/linux-yocto: additional workaround for qemuarm64-secureboot ZONE\_DMA

Reverting additional ZONE\_DMA patch 2281df0b0226610e235f49ed75bf6ad57cb04762 which was added in v5.10.37

Change-Id: I3700b95247a4224f4b56a403bc3dc10dde7dd5b5

Signed-off-by: Jon Mason <jon.mason@arm.com>

🕒 38 jobs for master in 55 minutes and 58 seconds (queued for 13 seconds)

📌latest

🔗2b887e62

📄

🔀No related merge requests found.

Pipeline

Needs

Jobs38

Failed Jobs3

Tests0

Prep

Bootstrap

Build

⚙️delete-dl-dir

⚙️delete-repo-...

⚙️delete-sstate

⚙️delete-toolc...

✅get-binary-to...

⚙️

✅machine-cov...

✅n1sdp/boots...

✅corstone500

✅corstone700-...

✅corstone700-...

❌fvp-base

❌fvp-base-ar...

❌

corstone500 (#1287932898) · Job x

← → ↺

gitlab.com/jonmason00/meta-arm/-/jobs/1287932898

GitLab

Projects Groups Snippets Help

Search or jump to...

Sign in / Register

M meta-arm

Project overview

Repository

Issues0

Merge requests0

Requirements

CI/CD

Pipelines

Jobs

Schedules

Test Cases

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Showing last 499.93 KiB of log - [Complete Raw](#)

4109 2021-05-24 08:46:13 - INFO - NOTE: Running task 3730 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-devtools/binutils/binutils\_2.36.bb:do\_packagedata)

4110 2021-05-24 08:46:13 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_packagedata: Started

4111 2021-05-24 08:46:14 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_packagedata: Succeeded

4112 2021-05-24 08:46:14 - INFO - NOTE: Running task 3731 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-devtools/binutils/binutils\_2.36.bb:do\_package\_qa)

4113 2021-05-24 08:46:14 - INFO - NOTE: Running task 3732 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-devtools/binutils/binutils\_2.36.bb:do\_package\_write\_ipk)

4114 2021-05-24 08:46:14 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_package\_qa: Started

4115 2021-05-24 08:46:14 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_package\_write\_ipk: Started

4116 2021-05-24 08:46:16 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_package\_qa: Succeeded

4117 2021-05-24 08:46:26 - INFO - NOTE: recipe perf-1.0-r9: task do\_install: Succeeded

4118 2021-05-24 08:46:26 - INFO - NOTE: Running task 3733 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_package)

4119 2021-05-24 08:46:26 - INFO - NOTE: Running task 3734 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_populate\_sysroot)

4120 2021-05-24 08:46:26 - INFO - NOTE: recipe perf-1.0-r9: task do\_package: Started

4121 2021-05-24 08:46:26 - INFO - NOTE: recipe perf-1.0-r9: task do\_populate\_sysroot: Started

4122 2021-05-24 08:46:27 - INFO - NOTE: recipe perf-1.0-r9: task do\_populate\_sysroot: Succeeded

4123 2021-05-24 08:46:28 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_package\_write\_ipk: Succeeded

4124 2021-05-24 08:46:28 - INFO - NOTE: Running task 3735 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-devtools/binutils/binutils\_2.36.bb:do\_rm\_work)

4125 2021-05-24 08:46:28 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_rm\_work: Started

4126 2021-05-24 08:46:29 - INFO - NOTE: recipe perf-1.0-r9: task do\_package: Succeeded

4127 2021-05-24 08:46:29 - INFO - NOTE: Running task 3736 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_packagedata)

4128 2021-05-24 08:46:29 - INFO - NOTE: recipe perf-1.0-r9: task do\_packagedata: Started

4129 2021-05-24 08:46:29 - INFO - NOTE: recipe perf-1.0-r9: task do\_packagedata: Succeeded

4130 2021-05-24 08:46:29 - INFO - NOTE: Running task 3737 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_package\_qa)

4131 2021-05-24 08:46:29 - INFO - NOTE: Running task 3738 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_package\_write\_ipk)

4132 2021-05-24 08:46:29 - INFO - NOTE: recipe perf-1.0-r9: task do\_package\_qa: Started

4133 2021-05-24 08:46:29 - INFO - NOTE: recipe perf-1.0-r9: task do\_package\_write\_ipk: Started

4134 2021-05-24 08:46:29 - INFO - NOTE: recipe binutils-2.36.1-r0: task do\_rm\_work: Succeeded

4135 2021-05-24 08:46:30 - INFO - NOTE: recipe perf-1.0-r9: task do\_package\_qa: Succeeded

4136 2021-05-24 08:46:31 - INFO - NOTE: recipe perf-1.0-r9: task do\_package\_write\_ipk: Succeeded

4137 2021-05-24 08:46:31 - INFO - NOTE: Running task 3739 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_rm\_work)

4138 2021-05-24 08:46:31 - INFO - NOTE: recipe perf-1.0-r9: task do\_rm\_work: Started

4139 2021-05-24 08:46:31 - INFO - NOTE: recipe perf-1.0-r9: task do\_rm\_work: Succeeded

4140 2021-05-24 08:46:31 - INFO - NOTE: Running noexec task 3740 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_rm\_work\_all)

4141 2021-05-24 08:46:31 - INFO - NOTE: Running noexec task 3741 of 3741 (/builds/jonmason00/meta-arm/work/poky/meta/recipes-kernel/perf/perf.bb:do\_build)

4142 2021-05-24 08:46:33 - INFO - NOTE: Tasks Summary: Attempted 3741 tasks of which 690 didn't need to be rerun and all succeeded.

4143 \$ ./ci/check-warnings \$KAS\_WORK\_DIR/build/warnings.log

4144 Cleaning up file based variables

4146 Job succeeded

corstone500

Duration: 32 minutes 57 seconds

Timeout: 4h (from project)

Runner: #6495233 (uBG7pJHP) arm64-meta-arm-runner-3

Commit 2b887e62

arm/linux-yocto: additional workaround for qemuarm64-secureboot ZONE\_DMA

✖ Pipeline #308354374 for master

build

⊕ fvp-base

⊕ fvp-base-arm32

⊕ fvp-base-arm32/external-gccar...

→ ✔ corstone500

✔ corstone700-fvp

✔ corstone700-mps3

00:00

⏴ Collapse sidebar

fv

p-base (#1287932914)

Jobs

gitlab.com/jonmason00/meta-arm/-/jobs/1287932914

Search or jump to...

Sign in / Register

M

meta-arm

Project overview

Repository

Issues0

Merge requests0

Requirements

CI/CD

Pipelines

Jobs

Schedules

Test Cases

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

2985 2021-05-24 08:14:48 - INFO - NOTE: recipe nettle-3.7.2-r0: task do\_populate\_lic: Succeeded

2986 2021-05-24 08:14:48 - INFO - NOTE: recipe psplash-0.14gitAUTOINC+0a902f7cd8-r15: task do\_deploy\_source\_date\_epoch: Succeeded

2987 2021-05-24 08:14:48 - INFO - NOTE: recipe busybox-1.33.1-r0: task do\_populate\_lic: Started

2988 2021-05-24 08:14:48 - INFO - NOTE: recipe libgcrypt-1.9.2-r0: task do\_patch: Started

2989 2021-05-24 08:14:48 - INFO - NOTE: recipe nettle-3.7.2-r0: task do\_deploy\_source\_date\_epoch: Succeeded

2990 2021-05-24 08:14:48 - INFO - NOTE: recipe busybox-1.33.1-r0: task do\_deploy\_source\_date\_epoch: Started

2991 2021-05-24 08:14:48 - INFO - NOTE: recipe file-5.40-r0: task do\_patch: Succeeded

2992 2021-05-24 08:14:48 - INFO - NOTE: recipe lzo-2.10-r0: task do\_patch: Succeeded

2993 2021-05-24 08:14:48 - INFO - NOTE: recipe acl-2.3.1-r0: task do\_patch: Succeeded

2994 2021-05-24 08:14:48 - INFO - NOTE: recipe busybox-1.33.1-r0: task do\_populate\_lic: Succeeded

2995 2021-05-24 08:14:48 - INFO - NOTE: recipe busybox-1.33.1-r0: task do\_deploy\_source\_date\_epoch: Succeeded

2996 2021-05-24 08:14:48 - INFO - NOTE: recipe ca-certificates-20210119-r0: task do\_install: Succeeded

2997 2021-05-24 08:14:49 - INFO - NOTE: recipe tcp-wrappers-7.6-r10: task do\_patch: Succeeded

2998 2021-05-24 08:14:49 - INFO - NOTE: recipe libgcrypt-1.9.2-r0: task do\_patch: Succeeded

2999 2021-05-24 08:14:51 - INFO - NOTE: Tasks Summary: Attempted 1345 tasks of which 656 didn't need to be rerun and 1 failed.

3000 2021-05-24 08:14:51 - INFO -

3001 2021-05-24 08:14:51 - INFO - Summary: 1 task failed:

3002 2021-05-24 08:14:51 - INFO - /builds/jonmason00/meta-arm/work/poky/meta/recipes-bsp/u-boot/u-boot\_2021.04.bb:do\_patch

3003 2021-05-24 08:14:51 - INFO - Summary: There were 8 WARNING messages shown.

3004 2021-05-24 08:14:51 - INFO - Summary: There was 1 ERROR message shown, returning a non-zero exit code.

3005 2021-05-24 08:14:51 - ERROR - Command "/builds/jonmason00/meta-arm/work/build\$ /builds/jonmason00/meta-arm/work/poky/bitbake/bin/bitbake -c build core-image-base perf" failed

3006 --- Error summary ---

3007 ERROR: u-boot-1\_2021.04-r0 do\_patch: Command Error: 'quilt --quiltrc /builds/jonmason00/meta-arm/work/build/tmp/work/fvp\_base-poky-linux/u-boot/1\_2021.04-r0/recipe-sysroot-native/etc/quiltrc push' exited with 0 Output:

3008 stdout: Applying patch 0001-vexpress64\_fvp-Set-DRAM-to-4G-to-match-with-kernel-d.patch

3009 patching file configs/vexpress\_aemv8a\_semi\_defconfig

3010 Hunk #1 FAILED at 2.

3011 1 out of 1 hunk FAILED -- rejects in file configs/vexpress\_aemv8a\_semi\_defconfig

3012 patching file include/configs/vexpress\_aemv8a.h

3013 Hunk #1 FAILED at 117.

3014 1 out of 1 hunk FAILED -- rejects in file include/configs/vexpress\_aemv8a.h

3015 Patch 0001-vexpress64\_fvp-Set-DRAM-to-4G-to-match-with-kernel-d.patch can be reverse-applied

3016 stderr:

3017 ERROR: Logfile of failure stored in: /builds/jonmason00/meta-arm/work/build/tmp/work/fvp\_base-poky-linux/u-boot/1\_2021.04-r0/temp/log.do\_patch.23852

3018 ERROR: Task (/builds/jonmason00/meta-arm/work/poky/meta/recipes-bsp/u-boot/u-boot\_2021.04.bb:do\_patch) failed with exit code '1'

3020 Cleaning up file based variables

3022 ERROR: Job failed: exit code 1

fv

p-base

Duration: 1 minute 44 seconds

Timeout: 4h (from project)

Runner: #6495232 (6-DCMAA5) arm64-meta-arm-runner-2

Commit 2b887e62

arm/linux-yocto: additional workaround for qemuarm64-secureboot ZONE\_DMA

Pipeline #308354374 for master

build

fv

fv

fv

corstone500

corstone700-fvp

corstone700-mps3

Collapse sidebar

00:03





qemu-cortex-m3/testimage [#12]

← → ↺

gitlab.com/jonmason00/meta-zephyr/-/jobs/1217484863

GitLab

Projects ▾ Groups ▾ More ▾

Search or jump to...

🔍 ↻ ⌵ ⌴

🔍 ↻ ⌵ ⌴

M

meta-zephyr

Project overview

Repository

Issues 0

Merge requests 0

Requirements

CI/CD

Pipelines

Editor

Jobs

Schedules

Test Cases

Security & Compliance

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Settings

49 2021-04-28 03:32:52 - INFO - cla50683225b0bf10f8d7a18280acefa7469ecc2

50 2021-04-28 03:32:52 - INFO - /builds/jonmason00/meta-zephyr/work/meta-openembedded\$ git checkout -q cla50683225b0bf10f8d7a18280acefa7469ecc2 -B dunfell

51 2021-04-28 03:32:52 - INFO - /builds/jonmason00/meta-zephyr/work/poky\$ git status -s

52 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/work/poky\$ git rev-parse --verify -q origin/dunfell

53 2021-04-28 03:32:53 - INFO - b0cb809fb9fe2828cf41ef1116952322c70a9e62

54 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/work/poky\$ git checkout -q b0cb809fb9fe2828cf41ef1116952322c70a9e62 -B dunfell

55 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/ci\$ git rev-parse --show-toplevel

56 2021-04-28 03:32:53 - INFO - Using /builds/jonmason00/meta-zephyr as root for repository meta-zephyr

57 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/ci\$ git rev-parse --show-toplevel

58 2021-04-28 03:32:53 - INFO - Using /builds/jonmason00/meta-zephyr as root for repository meta-zephyr

59 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/work/poky\$ /tmp/tmpjo\_l3q34/get\_bb\_env /builds/jonmason00/meta-zephyr/work/build

60 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/ci\$ git rev-parse --show-toplevel

61 2021-04-28 03:32:53 - INFO - Using /builds/jonmason00/meta-zephyr as root for repository meta-zephyr

62 2021-04-28 03:32:53 - INFO - /builds/jonmason00/meta-zephyr/work/build\$ /builds/jonmason00/meta-zephyr/work/poky/bitbake/bin/bitbake -c testimage zephyr-kernel-test-all

63 2021-04-28 03:36:23 - INFO - Parsing recipes...done.

64 2021-04-28 03:36:24 - INFO - Parsing of 1917 .bb files complete (0 cached, 1917 parsed). 2989 targets, 188 skipped, 0 masked, 0 errors.

65 2021-04-28 03:36:24 - INFO - NOTE: Resolving any missing task queue dependencies

66 2021-04-28 03:36:24 - INFO -

67 2021-04-28 03:36:24 - INFO - Build Configuration:

68 2021-04-28 03:36:24 - INFO - BB\_VERSION = "1.46.0"

69 2021-04-28 03:36:24 - INFO - BUILD\_SYS = "x86\_64-linux"

70 2021-04-28 03:36:24 - INFO - NATIVELSBSTRING = "debian-10"

71 2021-04-28 03:36:24 - INFO - TARGET\_SYS = "arm-yocto-eabi"

72 2021-04-28 03:36:24 - INFO - MACHINE = "qemu-cortex-m3"

73 2021-04-28 03:36:24 - INFO - DISTRO = "zephyr"

74 2021-04-28 03:36:24 - INFO - DISTRO\_VERSION = "1.0"

75 2021-04-28 03:36:24 - INFO - TUNE\_FEATURES = "armv7m vfp cortexm3"

76 2021-04-28 03:36:24 - INFO - TARGET\_FPU = "soft"

77 2021-04-28 03:36:24 - INFO - meta-zephyr = "HEAD:3bf98f9b97b53b59c7fc03502487ee7e766a037c"

78 2021-04-28 03:36:24 - INFO - meta-oe

79 2021-04-28 03:36:24 - INFO - meta-python = "dunfell:cla50683225b0bf10f8d7a18280acefa7469ecc2"

80 2021-04-28 03:36:24 - INFO - meta

81 2021-04-28 03:36:24 - INFO - meta-poky = "dunfell:b0cb809fb9fe2828cf41ef1116952322c70a9e62"

82 2021-04-28 03:36:24 - INFO -

83 2021-04-28 03:36:25 - INFO - Initialising tasks...done.

84 2021-04-28 03:36:25 - INFO - Sstate summary: Wanted 0 Found 0 Missed 0 Current 0 (0% match, 0% complete)

85 2021-04-28 03:36:25 - INFO - NOTE: No setscene tasks

86 2021-04-28 03:36:25 - INFO - NOTE: Executing Tasks

87 2021-04-28 03:36:25 - INFO - NOTE: Running task 1 of 1 (/builds/jonmason00/meta-zephyr/recipes-kernel/zephyr-kernel/zephyr-kernel-test-all.bb:do\_testimage)

88 2021-04-28 03:36:25 - INFO - NOTE: recipe zephyr-kernel-test-all-1.0-r0: task do\_testimage: Started

89 2021-04-28 03:36:25 - INFO - NOTE: recipe zephyr-kernel-test-all-1.0-r0: task do\_testimage: Succeeded

90 2021-04-28 03:36:26 - INFO - NOTE: Tasks Summary: Attempted 1 tasks of which 0 didn't need to be rerun and all succeeded.

92 Cleaning up file based variables

94 Job succeeded

00:01

qemu-cortex-m3/...

Duration: 3 minutes 46 seconds

Timeout: 2h (from project)

Commit 3bf98f9b

CI: add Gitlab CI support

Pipeline #293398421 for dunfell

test

→ qemu-cortex-m3/testimage

qemu-nios2/testimage

qemu-x86/testimage

# I'm sold, how do I set this up?

Gitlab Runner  
Gitlab Executor

# Gitlab Runner

A runner is a lightweight, highly-scalable agent that picks up a CI job through the coordinator API of GitLab CI/CD, runs the job, and sends the result back to the GitLab instance.

<https://docs.gitlab.com/ee/ci/runners/README.html>

# Gitlab Executor

- GitLab Runner implements a number of executors that can be used to run your builds in different scenarios
  - SSH
  - Shell
  - VirtualBox
  - Parallels
  - Docker
  - Kubernetes
  - Custom
- <https://docs.gitlab.com/runner/executors/README.html>

docs.gitlab.com/runner/executors/README.html

GitLab Docs

Search the docs...

GitLab.com (13.13-pre)

Get free trial

Issues

Merge requests

Operations

CI/CD

Application security

Packages & Registries

Infrastructure

Integrations

API

Omnibus GitLab

GitLab Runner

Install

Configure

Register

Executors

Custom

Docker

Docker Machine

Kubernetes

Shell

SSH

Parallels

Virtual Box

Monitor

Security

Shells

Troubleshoot

Best Practices

The list of executors above is locked. We no longer are developing or accepting new ones. Please check [CONTRIBUTION.md](#) to check the details.

## Selecting the executor

The executors support different platforms and methodologies for building a project. The table below shows the key facts for each executor which will help you decide which executor to use.

Executor	SSH	Shell	VirtualBox	Parallels	Docker	Kubernetes	Custom
Clean build environment for every build	X	X	✓	✓	✓	✓	conditional (4)
Reuse previous clone if it exists	✓	✓	X	X	✓	X	conditional (4)
Runner file system access protected (5)	✓	X	✓	✓	✓	✓	conditional
Migrate runner machine	X	X	partial	partial	✓	✓	✓
Zero-configuration support for concurrent builds	X	X (1)	✓	✓	✓	✓	conditional (4)
Complicated build environments	X	X (2)	✓ (3)	✓ (3)	✓	✓	✓
Debugging build problems	easy	easy	hard	hard	medium	medium	medium

1. It's possible, but in most cases it is problematic if the build uses services installed on the build machine

2. It requires to install all dependencies by hand

3. For example using [Vagrant](#)

4. Dependent on what kind of environment you are provisioning. It can be completely isolated or shared between each build.

5. When a runner's file system access is not protected, jobs can access the entire system including the runner's token, and the cache and code of other jobs. Executors marked ✓ don't allow the runner to access the file system by default. However, security flaws or certain configurations could allow jobs to break out of their container and access the file system hosting runner.

On this page

Selecting the executor

I am not sure

Shell executor

Virtual Machine executor (VirtualBox / Parallels)

Docker executor

Docker Machine executor

Kubernetes executor

SSH executor

Custom executor

Compatibility chart

Help and feedback

<< Collapse sidebar

CI/CD Settings · CI/CD · Settings

gitlab.com/jonmason00/meta-arm/-/settings/ci\_cd

GitLab

ProjectsGroupsMore

meta-arm

Project overview

Repository

Issues0

Merge requests0

Requirements

CI/CD

Security & Compliance

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Settings

General

Integrations

Webhooks

Repository

CI/CD

Operations

Pages

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. How do I configure runners?

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine. Runners are either:

active

 - Available to run jobs.

paused

 - Not available to run jobs.

Specific runners

These runners are specific to this project.

Set up a specific runner automatically

Register a runner on a Kubernetes cluster. [Learn more.](#)

1. Click the button below.

2. Select an existing Kubernetes cluster or create a new one.

3. From the Kubernetes cluster details view, applications list, install GitLab Runner.

Install GitLab Runner on Kubernetes

Set up a specific runner manually

1. Install GitLab Runner and ensure it's running.

2. Register the runner with this URL:

https://gitlab.com/

And this registration token:

h66mqF7Yw1bKJG2vYwFH

Reset registration token

Show Runner installation instructions

Available specific runners

#6585418 (KsY7zuXp)

internal-meta-arm-runner

aarch64arm64

Remove runner

#6495237 (9CmSzX6z)

arm64-meta-arm-runner-4

aarch64arm64

Remove runner

#6495233 (u8G7pjHP)

Remove runner

Shared runners

These runners are shared across this GitLab instance.

Shared Runners on GitLab.com

run in autoscale mode and are powered by Google Cloud Platform. Autoscaling means reduced wait times to spin up builds, and isolated VMs for each project, thus maximizing security.

They're free to use for public open source projects and limited to 400 CI minutes per month per group for private projects. Read about all [GitLab.com plans](#).

Enable shared runners for this project

Available shared runners: 15

#2568000 (z3WU8uu-)

shared-runners-manager-7

dockergo

#157328 (1d6b581d)

gitlab-shared-runners-manager-3.gitlab.com

gitlab-org

#2072991 (pVR9XBdq)

gitlab-docker-shared-runners-manager-04

gitlab-org-docker

#380986 (0277ea0f)

shared-runners-manager-5.gitlab.com

dockergo

#380990 (8a6210b8)

gitlab-shared-runners-manager-6.gitlab.com

gitlab-org

#2072964 (ih9XD9p3)

gitlab-docker-shared-runners-manager-03

gitlab-org-docker

Search or jump to...

12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012112212312412512612712812913013113213313413513613713813914014114214314414514614714814915015115215315415515615715815916016116216316416516616716816917017117217317417517617717817918018118218318418518618718818919019119219319419519619719819920020120220320420520620720820921021121221321421521621721821922022122222322422522622722822923023123223323423523623723823924024124224324424524624724824925025125225325425525625725825926026126226326426526626726826927027127227327427527627727827928028128228328428528628728828929029129229329429529629729829930030130230330430530630730830931031131231331431531631731831932032132232332432532632732832933033133233333433533633733833934034134234334434534634734834935035135235335435535635735835936036136236336436536636736836937037137237337437537637737837938038138238338438538638738838939039139239339439539639739839940040140240340440540640740840941041141241341441541641741841942042142242342442542642742842943043143243343443543643743843944044144244344444544644744844945045145245345445545645745845946046146246346446546646746846947047147247347447547647747847948048148248348448548648748848949049149249349449549649749849950050150250350450550650750850951051151251351451551651751851952052152252352452552652752852953053153253353453553653753853954054154254354454554654754854955055155255355455555655755855956056156256356456556656756856957057157257357457557657757857958058158258358458558658758858959059159259359459559659759859960060160260360460560660760860961061161261361461561661761861962062162262362462562662762862963063163263363463563663763863964064164264364464564664764864965065165265365465565665765865966066166266366466566666766866967067167267367467567667767867968068168268368468568668768868969069169269369469569669769869970070170270370470570670770870971071171271371471571671771871972072172272372472572672772872973073173273373473573673773873974074174274374474574674774874975075175275375475575675775875976076176276376476576676776876977077177277377477577677777877978078178278378478578678778878979079179279379479579679779879980080180280380480580680780880981081181281381481581681781881982082182282382482582682782882983083183283383483583683783883984084184284384484584684784884985085185285385485585685785885986086186286386486586686786886987087187287387487587687787887988088188288388488588688788888989089189289389489589689789889990090190290390490590690790890991091191291391491591691791891992092192292392492592692792892993093193293393493593693793893994094194294394494594694794894995095195295395495595695795895996096196296396496596696796896997097197297397497597697797897998098198298398498598698798898999099199299399499599699799899910001001100210031004100510061007100810091010101110121013101410151016101710181019102010211022102310241025102610271028102910301031103210331034103510361037103810391040104110421043104410451046104710481049105010511052105310541055105610571058105910601061106210631064106510661067106810691070107110721073107410751076107710781079108010811082108310841085108610871088108910901091109210931094109510961097109810991100110111021103110411051106110711081109111011111112111311141115111611171118111911201121112211231124112511261127112811291130113111321133113411351136113711381139114011411142114311441145114611471148114911501151115211531154115511561157115811591160116111621163116411651166116711681169117011711172117311741175117611771178117911801181118211831184118511861187118811891190119111921193119411951196119711981199120012011202120312041205120612071208120912101211121212131214121512161217121812191220122112221223122412251226122712281229123012311232123312341235123612371238123912401241124212431244124512461247124812491250125112521253125412551256125712581259126012611262126312641265126612671268126912701271127212731274127512761277127812791280128112821283128412851286128712881289129012911292129312941295129612971298129913001301130213031304130513061307130813091310131113121313131413151316131713181319132013211322132313241325132613271328132913301331133213331334133513361337133813391340134113421343134413451346134713481349135013511352135313541355135613571358135913601361136213631364136513661367136813691370137113721373137413751376137713781379138013811382138313841385138613871388138913901391139213931394139513961397139813991400140114021403140414051406140714081409141014111412141314141415141614171418141914201421142214231424142514261427142814291430143114321433143414351436143714381439144014411442144314441445144614471448144914501451145214531454145514561457145814591460146114621463146414651466146714681469147014711472147314741475147614771478147914801481148214831484148514861487148814891490149114921493149414951496149714981499150015011502150315041505150615071508150915101511151215131514151515161517151815191520152115221523152415251526152715281529153015311532153315341535153615371538153915401541154215431544154515461547154815491550155115521553155415551556155715581559156015611562156315641565156615671568156915701571157215731574157515761577157815791580158115821583158415851586158715881589159015911592159315941595159615971598159916001601160216031604160516061607160816091610161116121613161416151616161716181619162016211622162316241625162616271628162916301631163216331634163516361637163816391640164116421643164416451646164716481649165016511652165316541655165616571658165916601661166216631664166516661667166816691670167116721673167416751676167716781679168016811682168316841685168616871688168916901691169216931694169516961697169816991700170117021703170417051706170717081709171017111712171317141715171617171718171917201721172217231724172517261727172817291730173117321733173417351736173717381739174017411742174317441745174617471748174917501751175217531754175517561757175817591760176117621763176417651766176717681769177017711772177317741775177617771778177917801781178217831784178517861787178817891790179117921793179417951796179717981799180018011802180318041805180618071808180918101811181218131814181518161817181818191820182118221823182418251826182718281829183018311832183318341835183618371838183918401841184218431844184518461847184818491850185118521853185418551856185718581859186018611862186318641865186618671868186918701871187218731874187518761877187818791880188118821883188418851886188718881889189018911892189318941895189618971898189919001901190219031904190519061907190819091910191119121913191419151916191719181919192019211922192319241925192619271928192919301931193219331934193519361937193819391940194119421943194419451946194719481949195019511952195319541955195619571958195919601961196219631964196519661967196819691970197119721973197419751976197719781979198019811982198319841985198619871988198919901991199219931994199519961997199819992000200120022003200420052006200720082009201020112012201320142015201620172018201920202021202220232024202520262027202820292030203120322033203420352036203720382039204020412042204320442045204620472048204920502051205220532054205520562057205820592060206120622063206420652066206720682069207020712072207320742075207620772078207920802081208220832084208520862087208820892090209120922093209420952096209720982099210021012102210321042105210621072108210921102111211221132114211521162117211821192120212121222123212421252126212721282129213021312132213321342135213621372138213921402141214221432144214521462147214821492150215121522153215421552156215721582159216021612162216321642165216621672168216921702171217221732174217521762177217821792180218121822183218421852186218721882189219021912192219321942195219621972198219922002201220222032204220522062207220822092210221122122213221422152216221722182219222022212222222322242225222622272228222922302231223222332234223522362237223822392240224122422243224422452246224722482249225022512252225322542255225622572258225922602261226222632264226522662267226822692270227122722273227422752276227722782279228022812282228322842285228622872288228922902291229222932294229522962297229822992300230123022303230423052306230723082309231023112312231323142315231623172318231923202321232223232324232523262327232823292330233123322333233423352336233723382339234023412342234323442345234623472348234923502351235223532354235523562357235823592360236123622363236423652366236723682369237023712372237323742375237623772378237923802381238223832384238523862387238823892390239123922393239423952396239723982399240024012402240324042405240624072408240924102411241224132414241524162417241824192420242124222423242424252426242724282429243024312432243324342435243624372438243924402441244224432444244524462447244824492450245124522453245424552456245724582459246024612462246324642465246624672468246924702471247224732474247524762477247824792480248124822483248424852486248724882489249024912492249324942495249624972498249925002501250225032504250525062507250825092510251125122513251425152516251725182519252025212522252325242525252625272528252925302531253225332534253525362537253825392540254125422543254425452546254725482549255025512552255325542555255625572558255925602561256225632564256525662567256825692570257125722573257425752576257725782579258025812582258325842585258625872588258925902591259225932594259525962597259825992600260126022603260426052606260726082609261026112612261326142615261626172618261926202621262226232624262526262627262826292630263126322633263426352636263726382639264026412642264326442645264626472648264926502651265226532654265526562657265826592660266126622663266426652666266726682669267026712672267326742675267626772678267926802681268226832684268526862687268826892690269126922693269426952696269726982699270027012702270327042705270627072708270927102711271227132714271527162717271827192720272127222723272427252726272727282729273027312732273327342735273627372738273927402741274227432744274527462747274827492750275127522753275427552756275727582759276027612762276327642765276627672768276927702771277227732774277527762777277827792780278127822783278427852786278727882789279027912792279327942795279627972798279928002801280228032804280528062807280828092810281128122813281428152816281728182819282028212822282328242825282628272828282928302831283228332834283528362837283828392840284128422843284428452846284728482849285028512852285328542855285628572858285928602861286228632864286528662867286828692870287128722873287428752876287728782879288028812882288328842885288628872888288928902891289228932894289528962897289828992900290129022903290429052906290729082909291029112912291329142915291629172918291929202921292229232924292529262927292829292930293129322933293429352936293729382939294029412942294329442945294629472948294929502951295229532954295529562957295829592960296129622963296429652966296729682969297029712972297329742975297629772978297929802981298229832984298529862987298829892990299129922993299429952996299729982999300030013002300330043005300630073008300930103011301230133014301530163017301830193020302130223023302430253026302730283029303030313032303330343035303630373038303930403041304230433044304530463047304830493050305130523053305430553056305730583059306030613062306330643065306630673068306930703071307230733074307530763077307830793080308130823083308430853086308730883089

# How to get Gitlab CI building and running Yocto/OE

Gitlab CI config file

KAS

# Gitlab CI config file

Basically, a way to run shell scripts



image: ghcr.io/siemens/kas/kas

# First do a common bootstrap, and then build all the targets

```
stages:
  - prep
  - bootstrap
  - build
```

# Common job fragment to get a worker ready

```
.setup:
  stage: build
  variables:
    KAS_WORK_DIR: ${CI_PROJECT_DIR}/work
    KAS_REPO_REF_DIR: ${CI_BUILDS_DIR}/persist/repos
    SSTATE_DIR: ${CI_BUILDS_DIR}/persist/sstate
    DL_DIR: ${CI_BUILDS_DIR}/persist/downloads
    BB_LOGCONFIG: ${CI_PROJECT_DIR}/ci/logging.yml
    TOOLCHAIN_DIR: ${CI_BUILDS_DIR}/persist/toolchains
  before_script:
    - echo KAS_WORK_DIR = $KAS_WORK_DIR
    - echo SSTATE_DIR = $SSTATE_DIR
    - echo DL_DIR = $DL_DIR
    - mkdir --verbose --parents $KAS_WORK_DIR $KAS_REPO_REF_DIR $SSTATE_DIR $DL_DIR $TOOLCHAIN_DIR
```

# Generalised fragment to do a Kas build

```
.build:
  extends: .setup
  interruptible: true
  script:
    - KASFILES=$(./ci/jobs-to-kas ${CI_JOB_NAME})
    - kas shell --update --force-checkout $KASFILES -c 'cat conf/*.conf'
    - kas build $KASFILES
    - ./ci/check-warnings $KAS_WORK_DIR/build/warnings.log
```

#  
 # Prep stage, update repositories once  
 #

```
update-repos:
  extends: .setup
  stage: prep
  script:
    - flock --verbose --timeout 60 $KAS_REPO_REF_DIR ./ci/update-repos
```

```
get-binary-toolchains:
  extends: .setup
  stage: prep
  script:
    - ./ci/get-binary-toolchains $DL_DIR $TOOLCHAIN_DIR
```

#  
 # Bootstrap stage, bootstrap and machine coverage  
 #

```
# Build a number of native tools first to ensure the other builders don't race
# over them
nlsdp/bootstrap:
  extends: .build
  stage: bootstrap
```

# What percentage of machines in the layer do we build

```
machine-coverage:
  stage: bootstrap
```

# KAS

Tool provides an easy mechanism to setup bitbake based projects

- clone and checkout bitbake layers
- create default bitbake settings (machine, arch, ...)
- launch minimal build environment, reducing risk of host contamination
- initiate bitbake build process

Configuration via a yml file

```
header:
  version: 9

distro: poky

defaults:
  repos:
    refspec: master

repos:
  meta-arm:
    layers:
      meta-arm:
      meta-arm-bsp:
      meta-arm-toolchain:

poky:
  url: https://git.yoctoproject.org/git/poky
  layers:
    meta:
    meta-poky:

env:
  BB_LOGCONFIG: ""
  TOOLCHAIN_DIR: ""

local_conf_header:
  base: |
    CONF_VERSION = "1"
    PACKAGE_CLASSES = "package_ipk"
    LICENSE_FLAGS_WHITELIST += "armcompiler"
    PACKAGECONFIG_remove_pn-qemu-system-native = "gtk+ sdl"
    BB_NUMBER_THREADS = "16"
    PARALLEL_MAKE = "-j16"
    INHERIT += "rm_work"
    PACKAGECONFIG_append_pn-perf = " coresight"
    ERROR_QA = "${WARN_QA}"
  noptest: |
    DISTRO_FEATURES_remove = "ptest"

machine: unset

target:
  - core-image-base
  - perf
```

# KAS Docker

- Docker image created by the KAS project with all of the dependencies to build a Yocto image, as well as KAS installed

# Benefits of Developing this way

- Very reproducible builds
- Minimal environment and no dependency on installed packages
- Can use sstate and dl\_dir

# Cool features

Scheduling runs

Emails from failed/newly working runs

+

Pipeline Schedules - Jon Mason / x

←

→

↺

🔒

gitlab.com/jonmason00/meta-arm/-/pipeline\_schedules

☆

🔔

📺

🔌

🔍

👤

⋮

GitLab

ProjectsGroupsSnippetsHelp

Search or jump to...

Sign in / Register

M

meta-arm

🏠 Project overview

📁 Repository

📄 Issues0

🔀 Merge requests0

☰ Requirements

🔗 CI/CD

Pipelines

Jobs

Schedules

Test Cases

🔧 Operations

📦 Packages & Registries

📊 Analytics


📖 Wiki

🔗 Snippets

👥 Members

⏪ Collapse sidebar

Jon Mason > meta-arm > Schedules



Scheduling Pipelines

The pipelines schedule runs pipelines in the future, repeatedly, for specific branches or tags. Those scheduled pipelines will inherit limited project access based on their associated user.

Learn more in the [pipeline schedules documentation](#).

All4Active4Inactive0

Description	Target	Last Pipeline	Next Run	Owner
Nightly master	🔖 master	❌ #308354374	in 14 hours	👤 Jon Mason
Weekly hardknott	🔖 hardknott	✅ #308014717	in 5 days	👤 Jon Mason
Weekly gatesgarth	🔖 gatesgarth	✅ #308035538	in 5 days	👤 Jon Mason
Weekly dunfell	🔖 dunfell	✅ #308022903	in 5 days	👤 Jon Mason

[Archive](#) [Spam](#) [Delete](#) [Mark as unread](#) [Snooze](#) [Move to](#) [Labels](#) [More](#)

## Failed pipeline for master | meta-arm | 2b887e62

External

Inbox



GitLab &lt;gitlab@mg.gitlab.com&gt;

to

Sun, May 23, 4:37 PM (20 hours ago)



Reply



✖ Pipeline #308035656 has failed!

Project	Jon Mason / meta-arm
Branch	master
Commit	2b887e62 arm/linux-yocto: additional workaround for qemu...
Commit Author	Jon Mason

Pipeline #308035656 triggered by Jon Mason  
had 2 failed builds.

## Failed builds

✖ prep	<a href="#">update-repos</a>
✖ prep	<a href="#">get-binary-toolchains</a>



GitLab

You're receiving this email because of your account on [gitlab.com](#) · [Manage all notifications](#) · [Help](#)



# Debugging in Gitlab CI

Running docker locally to reproduce issue

```
jdm@athena:~$ sudo docker run -it --name kas-test --volume /yocto/downloads:/builds/persist/downloads --volume /yocto/sstate-cache:/builds/persist/sstate --volume /yocto/repos:/builds/persist/repos --volume /yocto/toolchains:/builds/persist/toolchains --privileged ghcr.io/siemens/kas/kas /bin/bash
builder@ea0827034b68:~$ git clone https://gitlab.com/jonmason00/meta-arm.git -b downstream
Cloning into 'meta-arm'...
remote: Enumerating objects: 579, done.
remote: Counting objects: 100% (579/579), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 7374 (delta 344), reused 529 (delta 310), pack-reused 6795
Receiving objects: 100% (7374/7374), 1.49 MiB | 7.77 MiB/s, done.
Resolving deltas: 100% (4204/4204), done.
builder@ea0827034b68:~$ cd meta-arm/
builder@ea0827034b68:~/meta-arm$ SSTATE_DIR=/builds/persist/sstate DL_DIR=/builds/persist/downloads kas build
ci/qemuarm64-secureboot.yml:ci/testimage.yml
```

# Potential Problems

Can't share sstate with non-container users using KAS

Difficult/impossible to access failing docker image

# Improvements?

Running in tmpfs

Runqemu working in non-privileged container

Optimal number of runners on a single host

# THANKS

---

<https://gitlab.com/jonmason00/meta-arm/-/pipelines>

To see more info on Gitlab CI running on meta-arm

CI/CD Settings · CI/CD · Settings

gitlab.com/jonmason00/meta-arm/-/settings/ci\_cd

GitLab

ProjectsGroupsMore

meta-arm

Project overview

Repository

Issues0

Merge requests0

Requirements

CI/CD

Security & Compliance

Operations

Packages & Registries

Analytics

Wiki

Snippets

Members

Settings

General

Integrations

Webhooks

Repository

CI/CD

Operations

Pages

Collapse sidebar

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. How do I configure runners?

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine. Runners are either:

active

 - Available to run jobs.

paused

 - Not available to run jobs.

Specific runners

These runners are specific to this project.

Set up a specific runner automatically

Register a runner on a Kubernetes cluster. [Learn more.](#)

1. Click the button below.

2. Select an existing Kubernetes cluster or create a new one.

3. From the Kubernetes cluster details view, applications list, install GitLab Runner.

Install GitLab Runner on Kubernetes

Set up a specific runner manually

1. Install GitLab Runner and ensure it's running.

2. Register the runner with this URL:

https://gitlab.com/

And this registration token:

h66mqF7Yw1bKJG2vYwFH

Reset registration token

Show Runner installation instructions

Shared runners

These runners are shared across this GitLab instance.

Shared Runners on GitLab.com

 run in **autoscale mode** and are powered by Google Cloud Platform. Autoscaling means reduced wait times to spin up builds, and isolated VMs for each project, thus maximizing security.

They're free to use for public open source projects and limited to 400 CI minutes per month per group for private projects. Read about all [GitLab.com plans](#).

Enable shared runners for this project

Available shared runners: 15

#2568000 (z3WU8uu-)

shared-runners-manager-7

docker

go

#157328 (1d6b581d)

gitlab-shared-runners-manager-3.gitlab.com

gitlab-org

#2072991 (pVR9XBdq)

gitlab-docker-shared-runners-manager-04

gitlab-org-docker

#380986 (0277ea0f)

shared-runners-manager-5.gitlab.com

docker

go

#380990 (8a6210b8)

gitlab-shared-runners-manager-6.gitlab.com

gitlab-org

#2072964 (ih9XD9p3)

gitlab-docker-shared-runners-manager-03

gitlab-org-docker

Available specific runners

#6585418 (KsY7zuXp)

internal-meta-arm-runner

aarch64

arm64

Remove runner

#6495237 (9CmSzX6z)

arm64-meta-arm-runner-4

aarch64

arm64

Remove runner

#6495233 (u8G7pjHP)

Remove runner

Collapse

# Gitlab Runner and executor docker from shell

- `$ sudo docker run -d --name gitlab-runner --restart always -v /var/run/docker.sock:/var/run/docker.sock gitlab/gitlab-runner:latest`
- `$ sudo docker exec -it gitlab-runner gitlab-runner register --non-interactive --tag-list "aarch64,arm64" --run-untagged --url "https://gitlab.com/" --registration-token "h6Gmqf7YwibKJG2vYwfH" --executor "docker" --docker-image "ghcr.io/siemens/kas/kas" --description internal-meta-arm-runner`